I J C T A, 8(5), 2015, pp. 2077-2083

A Hybrid Approach for Querying Numeric data in DAS-Model

Arjun K.R.* and Praveen K.**

Abstract: Database is the key component of any organization. Outsourcing these databases reduces the cost of managing and maintaining the database. Data in outsourced database are encrypted to provide privacy and confidentiality. In the proposed Hybrid model the data is initially bucketized using query optimal bucketization and then we apply an Order Preserving Encryption (OPE) on each bucket. The hybrid model combines bucketization and OPE. Since, we apply an OPE the comparison operation can be directly applied without decrypting the data. The OPE provides zero false positives with exact match. The proposed model preserves this zero false positives with increase in efficiency of querying of the database. Even if the attacker has access to the encrypted database he cannot decrypt the data without key and domain information.

Keywords: Database as Service, Order Preserving Encryption, Query Optimal Bucketization, False Positives, Performance Optimization.

Acronyms &Abbreviations: OPE– Order Preserving Encryption, DAS–Database as a service, QOB–Query Optimal Bucketization.

1. INTRODUCTION

Database as a service (DAS) model was first introduced by authors in 2002 [4], where client outsource their database to the trusted third party who performs maintenance and management of database. The client need not purchase any hardware/software and hire database administrator for maintaining the database. A major obstacle towards outsourcing the database is performance and insecurity of sensitive information [7]. In this DAS model the client data is stored at the service provider's premise. In order to protect privacy and confidentiality of the stored data the client encrypts the data and stores it to the database. In this paper we present a model which is designed to increase efficiency of processing the query. Bucketization is a technique for executing SQL queries over encrypted data. Data is partitioned and stored in different buckets. Depending on the bucketization technique the distribution of the data can be uniform or non-uniform. The proposed hybrid model can provide exact match of data with irrespective of query distribution. In DAS model there are different granularity levels of encrypting the data [4]. They are row, column, and cell level granularity. In this model we perform cell level encryption where each cell in the table is encrypted and stored. The OPE used can preserve the order of the encrypted data [1]. Following the normal encryption scheme results in less performance of query processing. For example, when a query is processed on normal encrypted column would lead to scan an entire table. Because the normal encryption scheme does not preserve the order and hence the database indices cannot be created which are used for searching [1]. Hybrid model combines the bucketization and order preserving encryption together which can outperform the existing DAS model in terms of efficiency.

The rest of the paper is organised as follows. Section 2 discuss about works related to database outsourcing and OPE. Section 3 discuss about Database as a service. Section 4 discuss about bucketization. Section 5

* TIFAC-CORE in Cyber Security, Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, Amrita University, India, *Email: arjun92kr@gmail.com*

** TIFAC-CORE in Cyber Security, Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, Amrita University, India, *Email: k_praveen@cb.amrita.edu*

discuss about Query Optimal Bucketization technique and algorithm. Section 6 discuss about Hybrid Model with example. Section 7 concludes the work and gives the further scope of research.

2. RELATED WORK

Database outsourcing uses trusted third party as the service provider and offers software, hardware and maintain the database. In order to protect the confidentiality of data, the organization encrypts the data before it is stored into database. Two main challenges in DAS model is data privacy and execution of SQL queries on encrypted database. Processing most of the queries at the server side without decryption can reduce overhead. Decryption of data and remainder of query processing can be done at client side. The author Hacigümüş et al. [3] proposed algebraic framework to split the query to minimize the computation at the client site. For querying an encrypted character string an n-phase reachability matrix for a string is defined and uses it as the characteristic index values.Theorems to split a SQL query into its server-side representation and client-side representation for partitioning the computation of a query across the client and the server is done and thus improving query performance [10]. Key management plays a critical role in encryption (not scope of this paper). There are different schemes of encryption. One such type of encryption is called Order Preserving Encryption. The encrypted data cannot be retrieved from the database using normal queries. Many algorithms have been developed to process the queries on encrypted data. The first order preserving encryption was proposed by authors in Agrawalet al[1]where the comparison operation can be applied directly on encrypted database. They have proved that the zero false positives exist. The new values can also be inserted without changing the encryption of other values. Even if the attacker gains access to the encrypted database he cannot decrypt the values without knowing the domain and key information used for encryption. The order preserving symmetric encryption was proposed in 2009 [2].The encryption scheme proposed is secure against the chosen plain text attacks which is not achieved using practical OPE. The key K is generated using pseudo random function which is used for encryption and decryption. The key is generated based on a random order-preserving function and the hyper geometric probability distribution [2]. Analysis of security in OPE was done by authors Xiao, Liangliang, and Yen, where the plaintext remains secret from the adversary against a known plaintext attack with known plaintexts. Previously the attacker can recover the plaintext P using a known plaintext attack. The adversary uses known plain text/cipher text pairs. The results prove that the attacker can retrieve some information about the plaintext P. The probability for the adversary to fully recover the plaintext P is a negligible function of log m if the number of known plaintexts/cipher text pairs satisfies $C = O(m^e)$, $0 < e < 1$ and $n \geq m^3$[11]. In this paper we present a Hybrid OPE which is more efficient in terms query processing.

3. DATABASE AS A SERVICE

Business organizations store their data in database. The amount of the data to be stored will be increasing over time and to managing and maintaining those data will be difficult task for the organization. To overcome these challenges organizations outsource their database to a trusted third party who stores and manages their database. These third party offer the service in less cost as compared to in-house data management cost. Since the service providers have access to the sensitive data of the organisation the problem of privacy arises. So the confidentiality of the data will be lost. In order to protect these data, the owner encrypts the data and stored in the database. Each record in the table is encrypted and stored in the outsourced database. Figure-1 is the architecture of DAS model [9].The DAS architecture consists of three modules: the user, the client, and server.

User: Queries the database requesting for data using normal query.

Client: Transforms the user query into a query which can be processed by encrypted database. Client module consists of query pre-processor, metadata, filter and result processor.

Server: Consists of encrypted database and query processor.

Initially 1.User submits the query to the database requesting the data from encrypted database. 2. Since the data stored in the server is in an encrypted form the normal query should be transformed into query as understandable by an encrypted database. Metadata is used to transform the query. The transformed query is submitted to server. 3. The server executes the query and retrieves results from database. These encrypted results are sent to client. 4. Client decrypts the results using the secret key. 5. Filter will communicate to query processor if any records are missing. After the decryption, plain text data is sent to user. The next section we will discuss some established techniques which are used to query encrypted database.

4. BUCKETIZATION

Bucketization is a technique to query an encrypted database for range queries which follow DAS model. The data is stored in the form of rows and columns in database. The records are encrypted and stored. The encrypted records are partitioned into several buckets and each bucket will have a bucket identifier (ID).

The range of data to be stored in each bucket is defined initially and only the client knows the information about each bucket like bucket id, minimum and maximum range of values in the corresponding bucket. The original queries are transformed into queries containing bucket identifier before transferring to the server. At the server side the query processor process the query and returns all records of a corresponding bucket. For example consider an Employee relation containing Salary, Age and Year as shown in Table 1. The data base consists of two buckets $B1$ and $B2$ where $B1$ contains employee details those who joined till 2010 and earlier and $B2$ contains the data of employees who joined after 2010.

When a client queries for details of an employee who joined after 2005. First step is to transform the query into a query containing bucket identifiers (using meta data). The transformed query is then submitted to the server and the server returns all the records matching the bucket identifier (Here in this Example entire bucket $B1$) to the client. At the client side the data is decrypted and sent to the user. Since the entire bucket is returned the number of false positives will be more (Here the number of false positives are 4). In order to reduce the numbers of false positives there are established bucketization algorithms which can reduce these false positives (Discussed in next section).

Normal Query: *SELECT * FROM Employee WHERE year > 2005*

Transformed Query: *SELECT * FROM Employee WHERE Bid = B1*

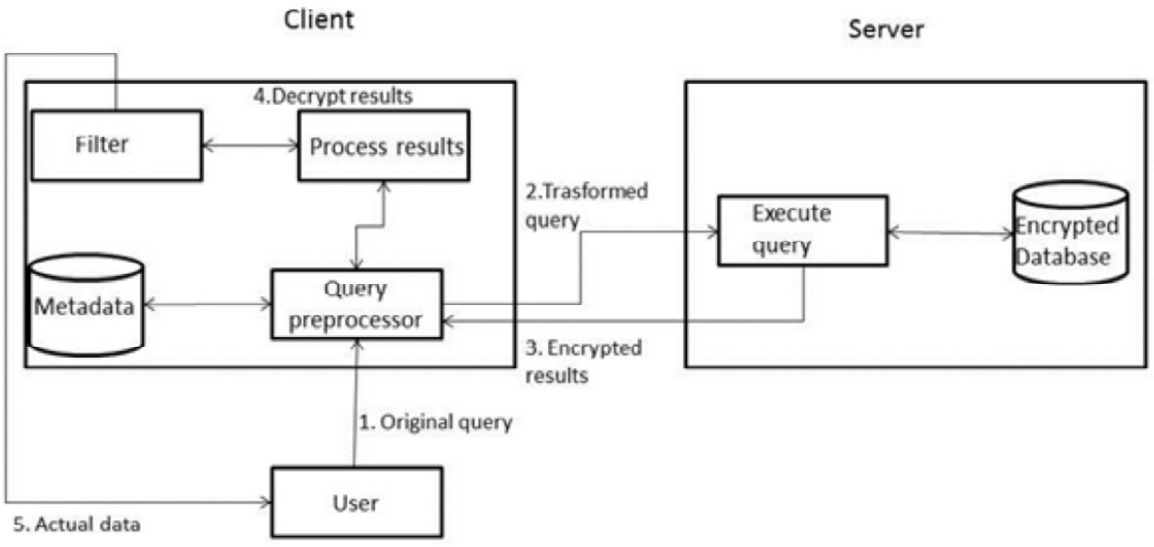

Figure 1: Architecture of DAS Model

Table 1
Employee Relation

	B1			*B2*	
Salary	*Age*	*Year*	*Salary*	*Age*	*Year*
10,000	32	1998	35,500	28	2011
15,000	25	2000	33,000	30	2011
20,000	40	2000	36,000	33	2012
22,000	43	2000	35,500	35	2013
25,000	46	2006	35,500	42	2013
32,000	37	2009	40,000	45	2015
35,000	42	2010	42,000	44	2015

5. QUERY OPTIMAL BUCKETIZATION

Query optimal bucketization is a technique to minimize the false positives and reduce the bucket cost for each bucket. QOB provides optimal solution to bucketize a set of values, $V = \{v_1, v_2, \ldots\ldots, v_n\}$, for M buckets. All the values in set V should be sorted i.e, $v_1 < \ldots\ldots < v_n$, where each value should present at least once in the table [6]. The problem of bucketization follows optimal sub-structure property. The optimal solution can be achieved by dividing the problem into two smaller sub-problems where one contains the leftmost *M-1* buckets covering *(n-i)* smallest points from V and other containing the extreme right single bucket covering the remaining largest i points from V [6].

$$QOB(1, n, M) = Min_i[QOB(1, n - i, M - 1) + BC(n - i + 1, n)] \quad (1)$$

Each bucket contains the range of values $[v_i, v_j]$, where bucket cost BC is given by

$$BC(i, j) = (v_j - v_i + 1) * \sum_{i<t<j} f_t \quad (2)$$

The algorithm accepts input as data set consists of values and frequency of each value. Output of the algorithm is a minimum cost for bucketing the data set and the partition point[1].

ALGORITHM 1.QOB(*D, M*)

Input: Data set D= (*Value, Frequency*) and bucket size *M*
(Where |Value| = |Frequency| = n)

Output: Optimal bucket cost& matrix H

Initialize

(i) Matrix $H[n][M]$ to 0

(ii) Matrix $OPP[n][M]$ to 0

For $k = 1 \ldots\ldots n$ //For sub-problem with bucket size $M = 2$

$H[k][2] = \mathbf{Min}_{2 \le i \le k-1}$ (BC(1, *i*) + BC(*i* + 1, *k*))

Store optimal-partition-point ibest in $OPP[k][2]$

For $l = 3 \ldots\ldots M$ //Combining sub problems

For $k = 1 \ldots\ldots n$

$H[k][l] = \mathbf{Min}_{(l-1 \le i \le k-1)}$ (H[*i*] [*l*-1] + BC(*i* + 1, *k*))

Store optimal-partition-point ibest in $OPP[k][l]$

OUTPUT: "Minimum Bucketization Cost = $H[n][M]$"

6. HYBRID MODEL (BUCKETIZATION + OPE)

In this section we will describe our conceptual model which integrates bucketization and Order Preserving Encryption. The proposed system works in two phases: I) Initially the data is bucketized using QOB algorithm

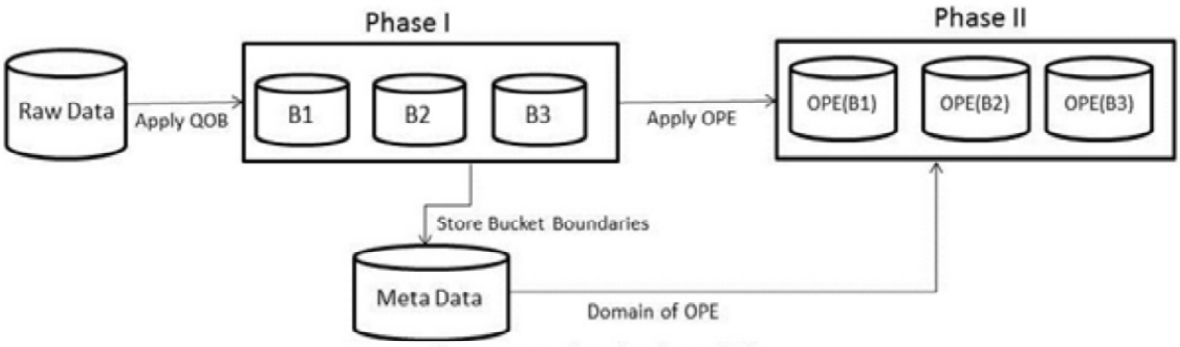

Figure 2: Architecture of Hybrid Model

which calculates partitions points and minimum cost of bucketization. II) Apply an order preserving encryption scheme on each bucket. We perform bucketization as followed in QOB algorithm which is described in previous section.

The number of false positives will be reduced since the data is partitioned using QOB bucketization. In the given employee relation we try to bucketize the data using attribute YEAR (Table 1). We partition the data using QOB algorithm with considering the size of bucket $M = 3$. The input to the algorithm is attribute value and frequency[2]. At the end of phase-I bucket B1 contains data corresponding to Year {1998, 2000}, B2 containing {2006, 2009, 2010}and B3 containing {2011, 2012, 2013, 2015}. Table 3 shows the data stored in buckets $B1$ $B2$.

In the phase II, we apply an OPE algorithm and store the data (Table 3) into database. When we apply an OPE the numerical order of the plaintext is preserved even after the encryption. A major disadvantage of encrypting sensitive data is that the data needs to be deciphered for query processing. An Order preserving symmetric encryption scheme produces the cipher text that preserves the numerical ordering of the plaintexts [1]. A general OPE scheme with plaintext-space [P] and cipher text space [C] is defined, OPE = (RK, ENC, DEC) where RK is the random key generated using randomized key-generation algorithm, ENC and DEC are encryption and decryption algorithm which uses RK as the key

$$DEC(ENC\ (RK, m)) = m$$

for every plaintext value $\{m1, m2\}$ in P and $\{c1, c2\}$in C with key RK, the OPE property holds

Table 2
Frequency Table

Index No	*Year*	*Frequency*
1	1998	1
2	2000	3
3	2006	1
4	2009	1
5	2010	1
6	2011	2
7	2012	1
8	2013	2
9	2015	2

This scheme reveals no information about the original values apart from their order. The adversary will not know about the plain text values without mapping of plain text and cypher text. Table 4 shows that the order of the data is preserved even after the data is encrypted. This scheme supports both point and rangequeries (MIN, MAX, COUNT).

Table 3
Data to be stored in buckets B1 and B2 after Phase II

	B1			*B2*	
Salary	*Age*	*Year*	*Salary*	*Age*	*Year*
10,000	32	1998	25,000	46	2006
15,000	25	2000	32,000	37	2009
20,000	40	2000	35,000	42	2010
22,000	43	2000			

Many queries can be directly processed without decrypting the data. Few queries which include SUM, AVG requires the decryption of data. Each bucket will be encrypted using a different key RK through which the security can be enhanced. The Key and the corresponding bucket identifier will be stored in the meta data of the client. Consider a query which requests for details of employees joined between 1999 and 2001. The query will be of the form

SELECT FROM employee WHER Eyear > 1999 AND year < 2001*

This query is transformed into a query containing bucket identifier and the values (1999 & 2001) are mapped into a cipher text using the key stored in Meta data. So the transformed query will be of the form

*SELECT * FROM employee WHERE Bid = B*1 *AND year > 200 AND year* < 215

Table 4
Bucket B1-After applying OPE

	B1	
Salary'	*Age'*	*Year'*
1500	20	198
2000	15	210
2500	30	210
2300	35	210

The query results contain no false positives (exact match). Any updates or insertions to the table can be done without affecting the order of records already stored. Some of the queries do not require the decryption (COUNT). Queries which include MIN, MAX will require only decryption of one record because the order is preserved.

7. CONCLUSION& FUTURE WORK

In this paper we proposed a model of storing and querying a numerical database. It is proved that zero false positives exits in the OPE scheme[1, 2]. In this proposed hybrid model we can assure that the queries can be processed efficiently with the help of QOB bucketization. With the help of bucketization the search space for the data retrieval is minimized to fewer buckets. Instead of searching and decrypting the entire table the queries corresponding to the particular buckets are decrypted. Hence the efficiency will be more compared to normal encrypted databases. Future work of this paper will be on security analysis of order preserving schemes followed. Each bucket can be encrypted using different key but overhead of meta data should be considered. Query access pattern which leads to inference attacks should also be considered while choosing OPE schemes[5].

Notes

1. The algorithm is referred from [6] and an input example is also provided.
2. Input is the (Value, Frequency)={1998,1; 2000,3; 2006,1; 2009,1; 2010,1; 2011,2; 2012,1; 2013,2; 2015,2}. The output obtained is 2, 5, 9 as bucket boundaries. These bucket boundaries are stored as Meta data in the client side.

Reference

[1] Agrawal, Rakesh, et al. "Order preserving encryption for numeric data." Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004.

[2] Boldyreva, Alexandra, Nathan Chenette, and Adam ONeill. "Order-preserving encryption revisited: Improved security analysis and alternative solutions." Advances in Cryptology CRYPTO 2011. Springer Berlin Heidelberg, 2011. 578-595.

[3] Hacigümüş, Hakan, et al. "Executing SQL over encrypted data in the database-service-provider model." Proceedings of the 2002 ACM SIGMOD international conference on Management of data. ACM, 2002.

[4] Hacigümüs, Hakan, BalaIyer, and SharadMehrotra."Providing database as a service." Data Engineering, 2002. Proceedings. 18th International Conference on. IEEE, 2002.

[5] Hore, Bijit, et al. "Secure multidimensional range queries over outsourced data." The VLDB Journal-The International Journal on Very Large Data Bases 21.3 (2012): 333-358.

[6] Hore, Bijit, SharadMehrotra, and Gene Tsudik. "A privacy-preserving index for range queries." Proceedings of the Thirtieth international conference on Very large databases-Volume 30. VLDB Endowment, 2004.

[7] Jhawar, Ravi, Vincenzo Piuri, and Pierangela Samarati."Supporting security requirements for resource management in cloud computing." Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on. IEEE, 2012.

[8] Mykletun, Einar, and Gene Tsudik. "Aggregation queries in the database-as-a service model." Data and Applications Security XX. Springer Berlin Heidelb erg, 2006. 89-103.

[9] Raybourn, Tracey. Bucketization Techniques for Encrypted Databases: Quantifying the Impact of Query Distributions. Diss. Bowling Green State University, 2013.

[10] Wu, ZongDa, et al. "Executing SQL queries over encrypted character strings in the Database-As-Service model." Knowledge-Based Systems 35 (2012): 332-348.

[11] Xiao, Liangliang, and I. Yen. "Security analysis for order preserving encryption schemes." Information Sciences and Systems (CISS), 2012 46th Annual Conference on. IEEE, 2012.